

1/12

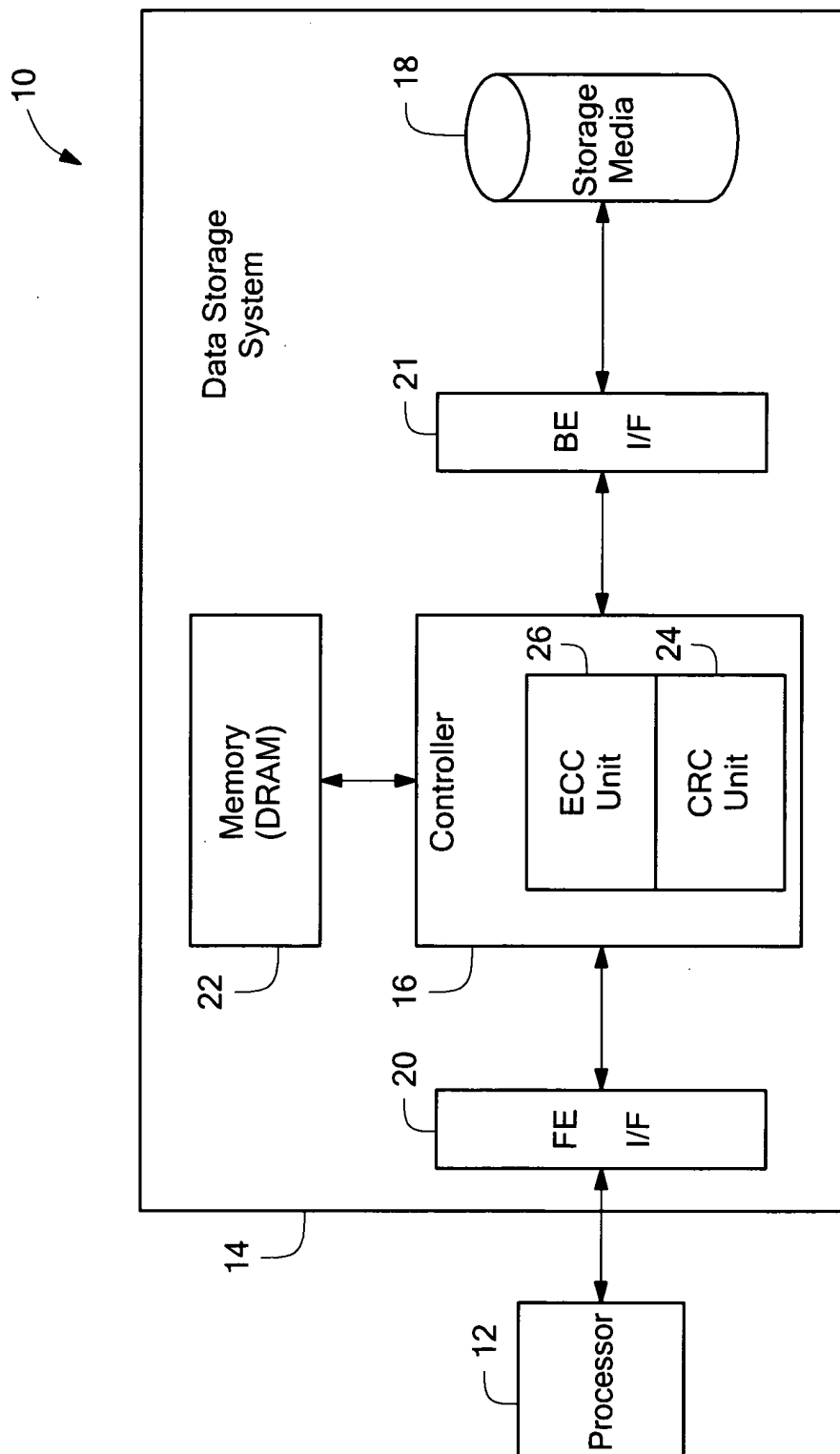


FIG. 1

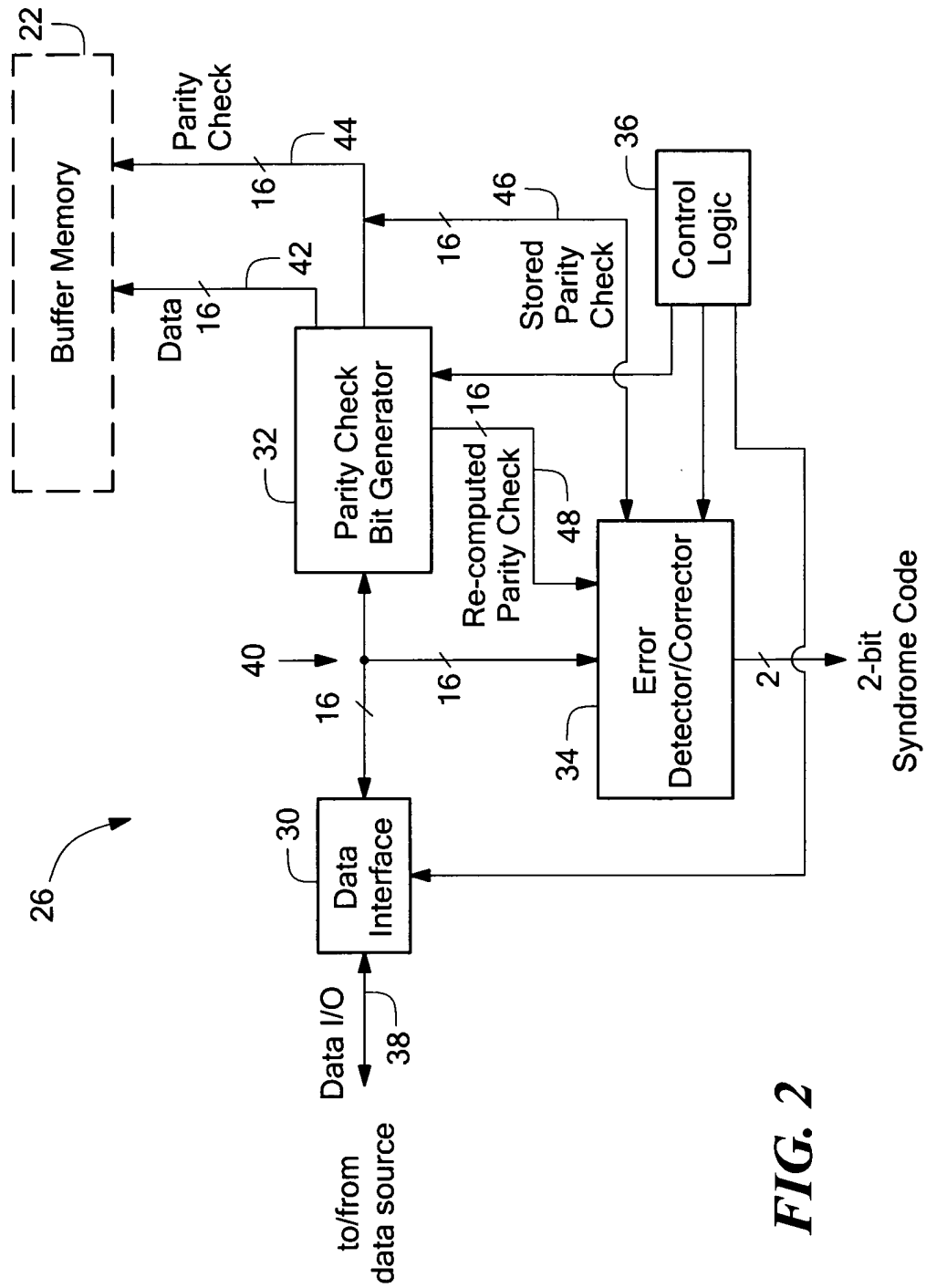
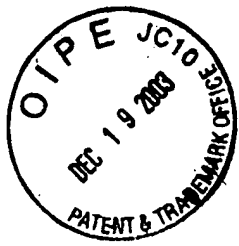
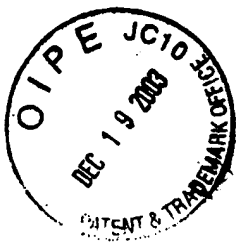


FIG. 2



SIMPLE ERROR-CORRECTION CODES FOR DATA BUFFERS

Lih-Jyh Weng et al.

10/608,320

3/12

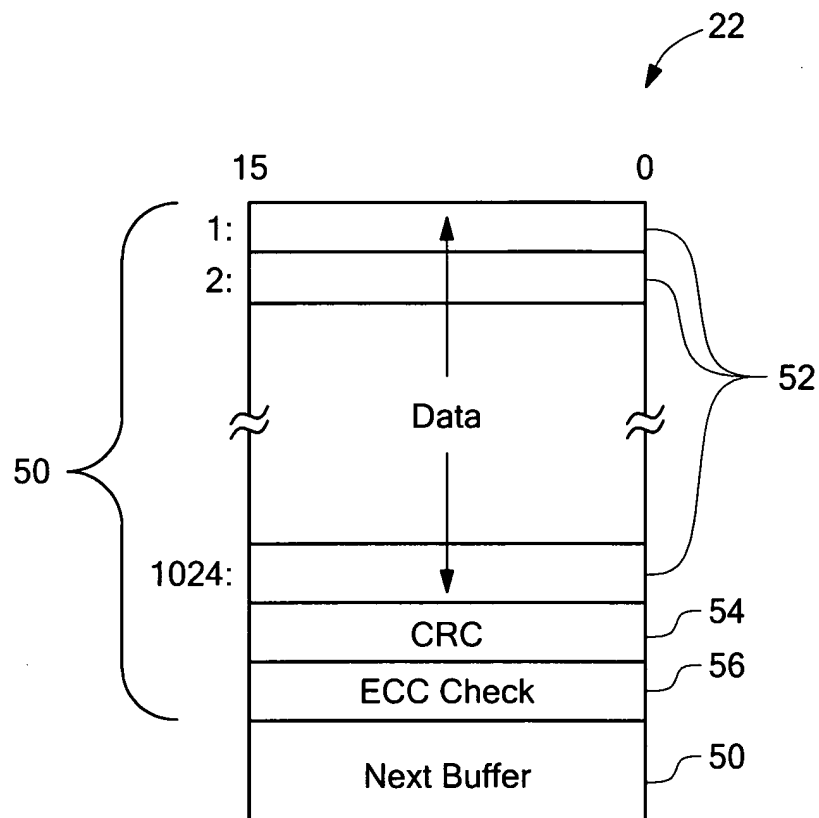
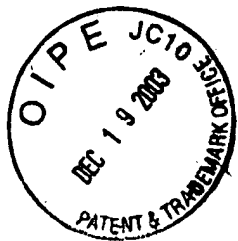


FIG. 3

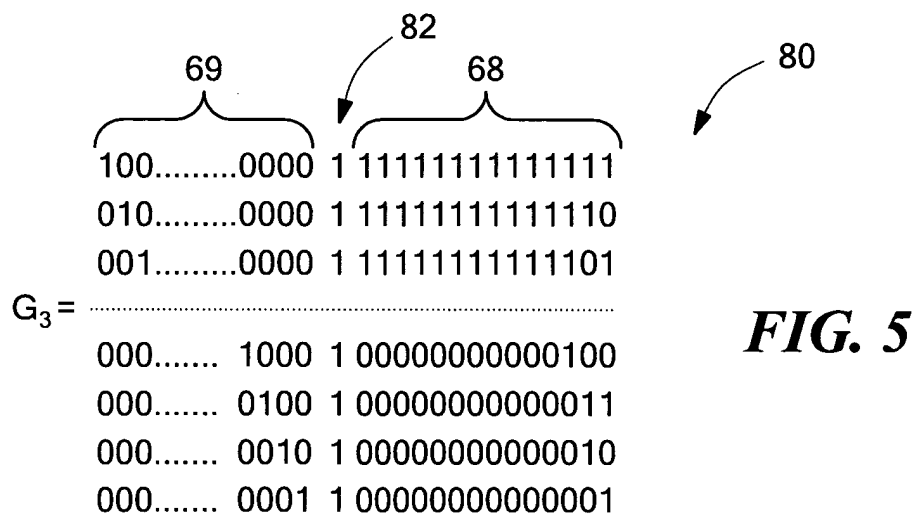
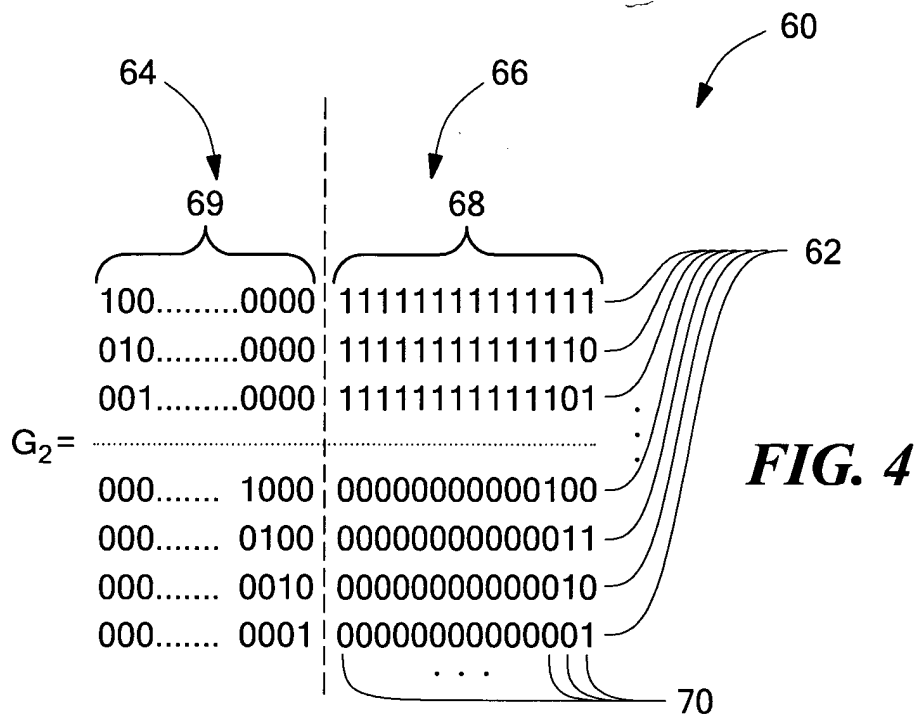


SIMPLE ERROR-CORRECTION CODES FOR DATA BUFFERS

Lih-Jyh Weng et al.

10/608,320

4/12



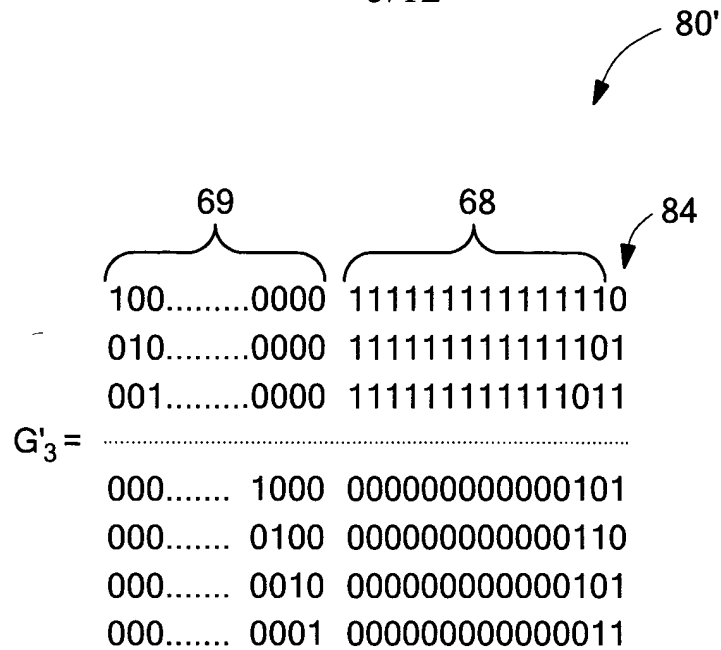
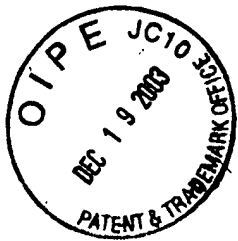


FIG. 6

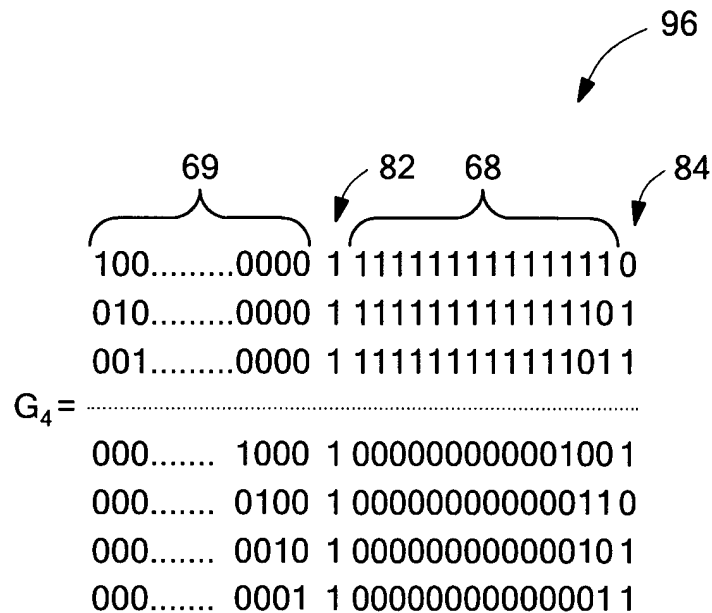
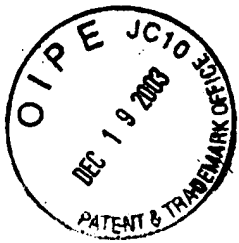


FIG. 7



SIMPLE ERROR-CORRECTION CODES FOR DATA BUFFERS

Lih-Jyh Weng et al.

10/608,320

6/12

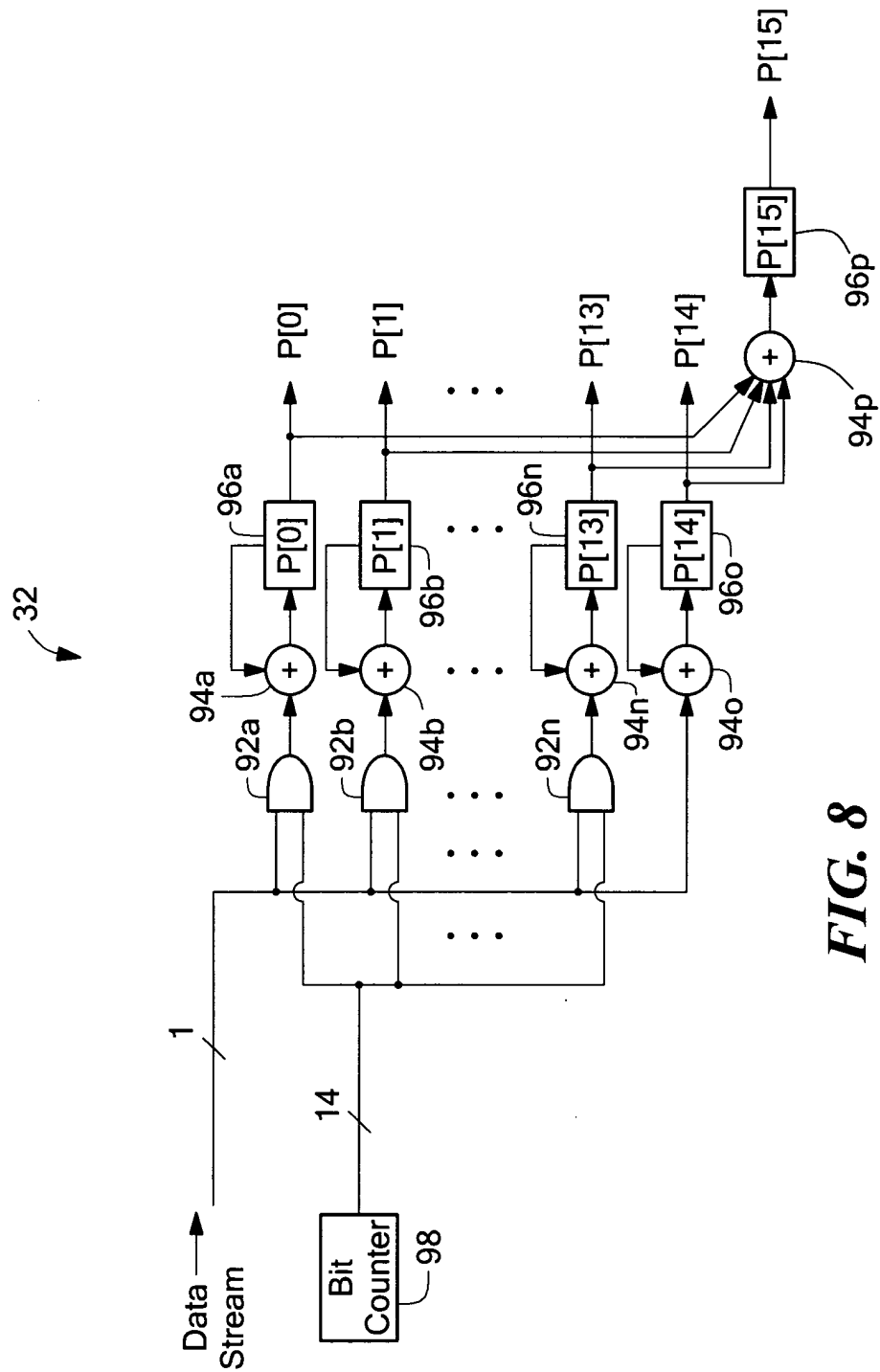
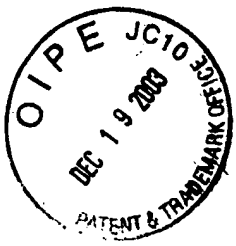


FIG. 8



SIMPLE ERROR-CORRECTION CODES FOR DATA BUFFERS

Lih-Jyh Weng et al.

10/608,320

7/12

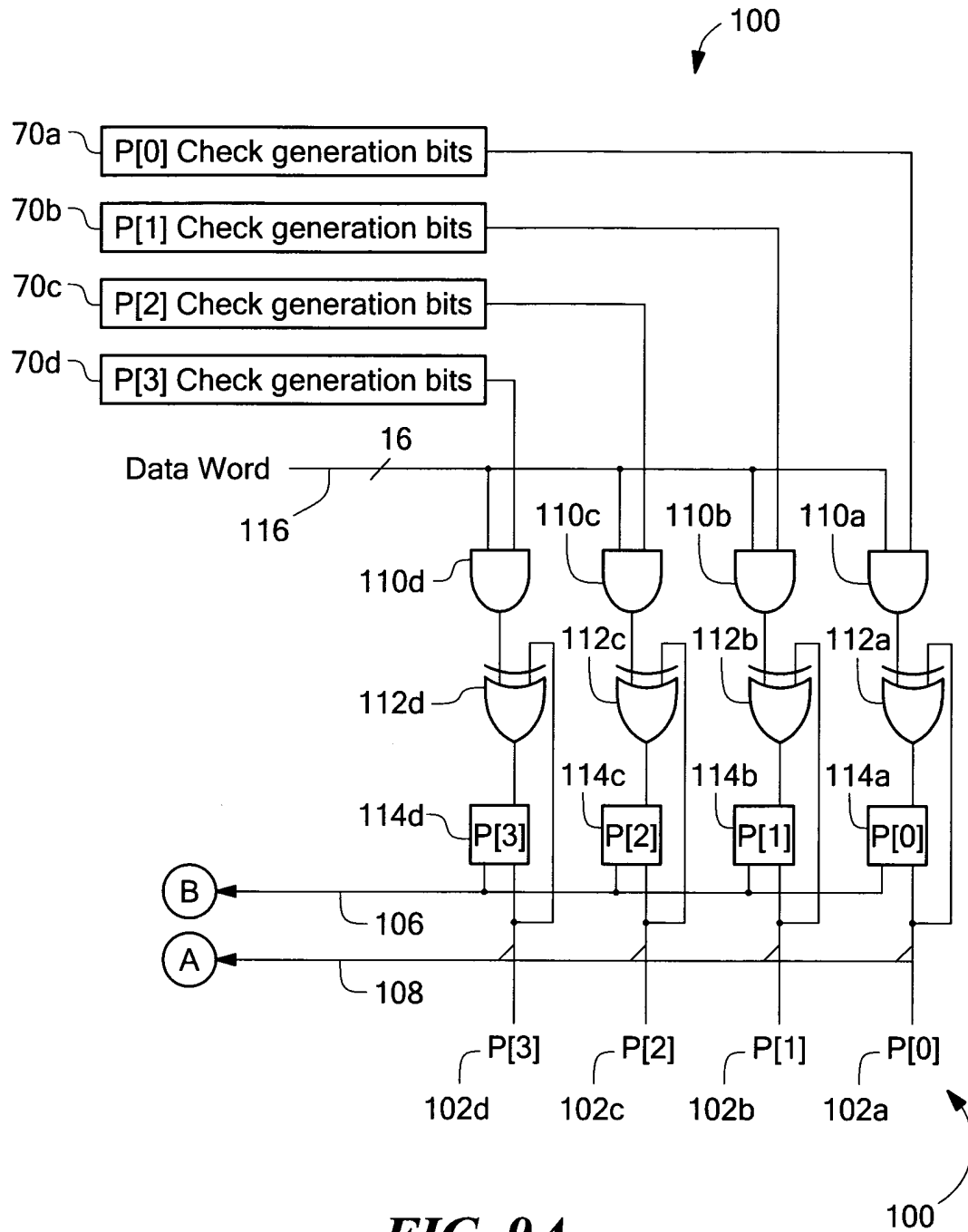


FIG. 9A



SIMPLE ERROR-CORRECTION CODES FOR DATA BUFFERS

Lih-Jyh Weng et al.

10/608,320

9/12

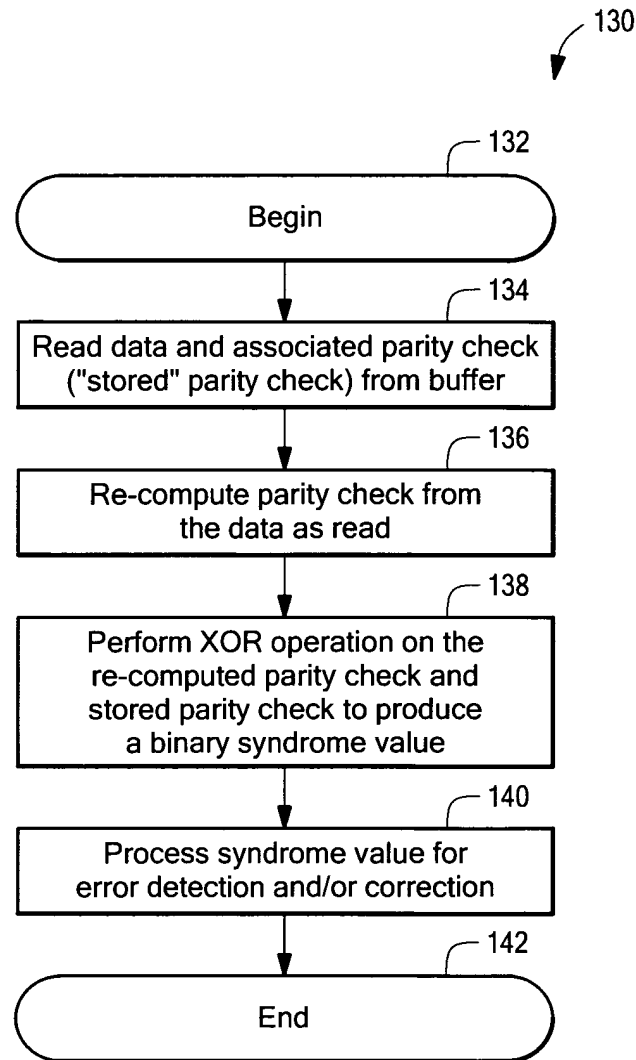


FIG. 10

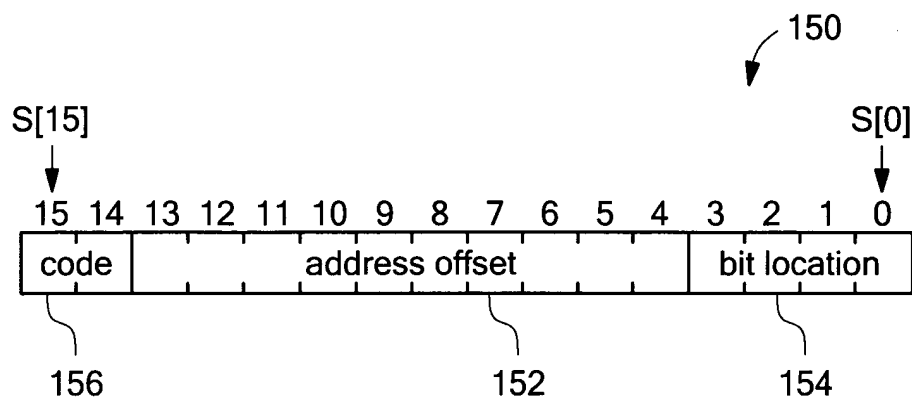
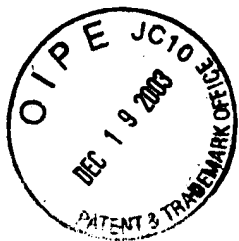


FIG. 11

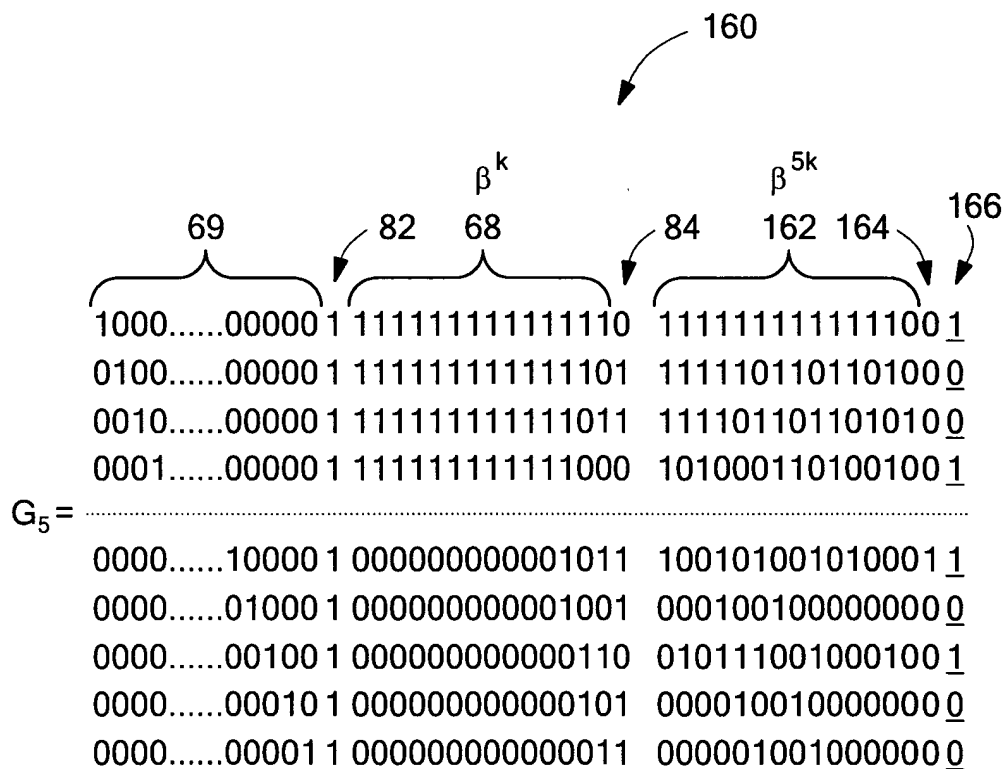
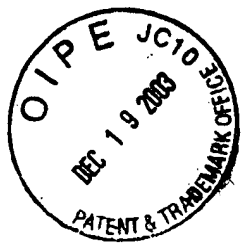


FIG. 12



SIMPLE ERROR-CORRECTION CODES FOR DATA BUFFERS

Lih-Jyh Weng et al.

10/608,320

unsigned encode()

11/12

170

```
{    Int M = 14;

    unsigned result;

    unsigned syndrome_x_0, syndrome_x_1, syndrome_x_5;

    int i,j;

    unsigned parity_on_parity;

    unsigned position; // The value of position starts at 1 and ends at 2^14 - 1.

    syndrome_x_0 = 0;

    syndrome_x_1 = 0;

    syndrome_x_5 = 0;

    for(i=0; i < CODE_LENGTH_IN_LONG_WORD; i++) {
        for (j = 0; j < 32; j++)    {
            if(p_temp[i] & (1<<j)) {
                syndrome_x_0^= 1;
                position = (unsigned) i*32 + j + 1;
                syndrome_x_1^= add_parity(position);
                syndrome_x_5^= add_parity(fifth_power(position));
            }
        }
    }

    parity_on_parity = syndrome_x_0^ (syndrome_x_1 & 1) ^ (syndrome_x_5 & 1);
```

result = 0;

result ^= syndrome_x_5 << (30-M); // bits 16 - 30

result ^= syndrome_x_1 << (29-M-M); // bits 1 - 15

result ^= syndrome_x_0 << 1; // bit 31

result ^= parity_on_parity; // bit 0

return(result);

}

FIG. 13

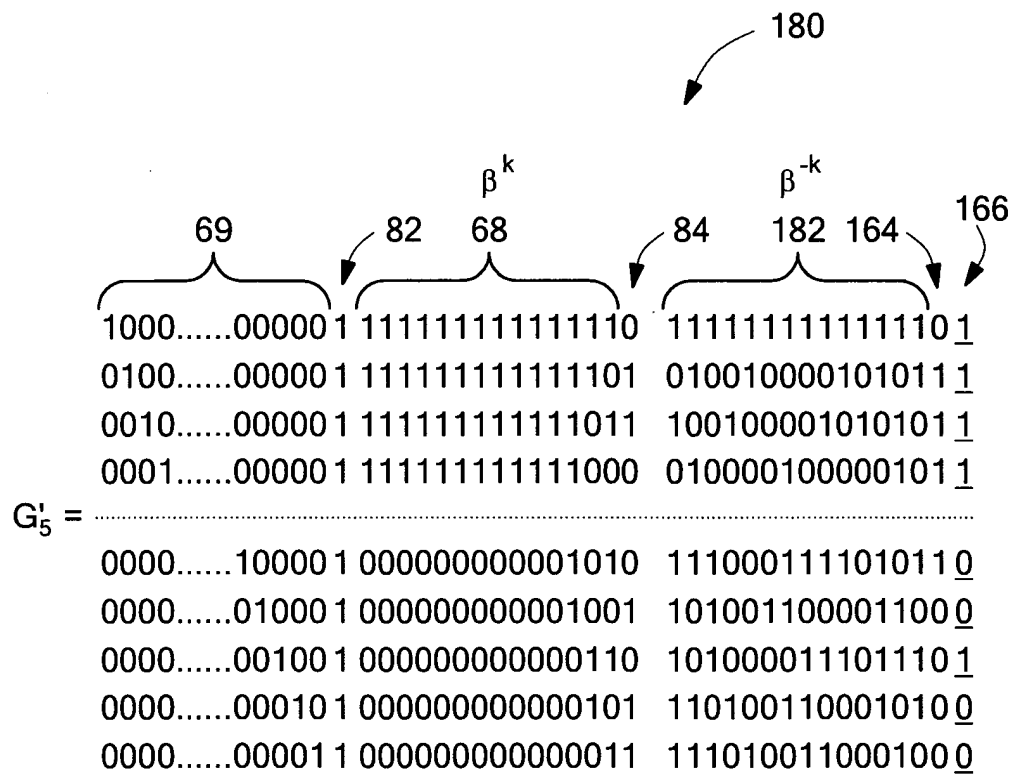


FIG. 14